Shared Semantic Representations for Coordinating Distributed Robot Teams

Dac Le and Ian Horswill
Computer Science Department / The Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue
Evanston, IL 60201
Email: ian@cs.nwu.edu
http://www.cs.nwu.edu/~ian

Abstract

Most multi-robot systems are limited to simple reactive architectures (Mataric 97, Parker 94, Bekey et al. 96). These architectures interface cleanly to sensors and effectors, support limited incremental development, and can support real-time performance on low cost compute hardware (Brooks 86). However, they limit designs to simplistic representations, such as numeric signals. These limitations—

- Hinder design reuse by making it difficult to pass parameters to components; collecting green pucks and collecting red pucks are often implemented by separate components
- Hinder robot-computer interfacing because humans typically don't think and talk at the level of sonar readings
- Severely limit the space of tasks that can be performed *at all*.; they cannot tractibly implement human/robot dialog, for example

We believe that by providing support for *structured representations* and *simple cognitive processing* in multi-robot systems, we can greatly improve component reusability, allow humans to direct robot teams with useful high level instructions, and solve dramatically more complex tasks. We have developed a class of single-agent architectures, collectively referred to as *role-passing architectures*, that support forward- and backward-chaining inference, means-ends analysis, and reasoning about the agent's current knowledge and goals. Unlike conventional symbolic AI systems, however, they provide all of the performance characteristics of behavior-based systems [Arkin 98]. In particular, the architecture--

- Supports distributed representation
- Provides hard real-time guarantees
- Maps well into both coarse- and fine-grained parallel architectures, as well as conventional serial architectures
- Interfaces simply and cleanly with sensors and effectors
- Can be implemented using low cost, low power components
- Can be directly implemented in silicon (e.g. using FPGAs) for tasks requiring super-human performance (i.e. billions of inferences per second)

Our current implementation [Horswill 98; Beim et al. 97], running on two robots, Hack and Kludge, supports--

- Simple natural language instruction following in the domain of fetch-and-follow tasks using a 30 word vocabulary
- Reactive problem solving involving inference, subgoal hierarchies, and explicit reasoning about epistemic actions
- Goal-directed visual search and adaptive color-based tracking of up to three deformable objects (Horswill and Barnhart 95)
- Vision-based navigation using the Polly algorithm (Horswill 95)

simultaneously, all in real time, and all on a low cost 25 MIP DSP. While the current implementation is interpreted, it can be easily compiled for higher performance and exceptionally good scaling properties. We estimate that a 1000 rule inference network could be run at 1000 Hz (i.e. 1000 complete reevaluations of the knowledge base per second) on a StrongARM microcontroller with less than 65K of RAM. Such a system would require roughly 250mW.

As part of the the DARPA-sponsored Distributed Robotics program, we are now working to extend this technology to multi-robot systems by linking team members with a wireless network to exchange sensor and inference data. Since the architecture already uses distributed processing and implements inference with a feed-forward combinational logic network, it is straight-forward to further distribute computation across robots. This would provide team members with a shared situation assessment and limited on-demand access to the sensory data of other team members. In effect, they would have a kind of "group mind." We believe such an architecture would enable:

- **Improved situational awareness**
  Because the architecture completely recomputes all inferences at sensor rates, it can respond to contingencies as soon as they are sensed. Because team members will be linked, the team can respond as a whole as soon as a member detects a problem.
- **Improved team coordination**
  Team members can coordinate using high level descriptions of the situation [Tambe 97]. The same mechanisms that are used to track and fuse data within an individual can be used to fuse data across individuals.
- **Improved human-robot coordination**
  Human commanders will be able to monitor the team by monitoring the network traffic using a high level multimodal interface. Commands can be given to the team by injecting additional requests into the network. Since much of the network traffic consists of structured representations, it is already at the right level of abstraction for human communication. We have developed prototype systems for parsing and executing simple natural language commands on single-robot systems, extending the technology to multi-robot systems should be straight-forward

References

[Arkin 98] Ron Arkin. *Behavior-based Robots*. MIT Press, 1998.

[Beim et al. 97] CPT. Peter Beim, I. Horswill, and I. Yen. "Hack and Kluge." In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI. July 1997.

[Brooks 86] Rodney Brooks. "A Robust Layered Control System For a Mobile Robot." *IEEE Journal of Robotics and Automation* 2:1 (1986) pp. 14-23.

[Horswill 95] Ian Horswill. "Analysis of Adaptation and Environment." *Artificial Intelligence* 73(1-2) (1995) 1-30. Special double-volume on Computational Theories of Interaction and Agency, Rosenschein and Agre, eds.

[Horswill 98] Ian Horswill. "Grounding Mundane Reasoning in Perception." *Autonomous Robots* 5:1. Kluwer, February 1998.

[Mataric 97] Maja Mataric. "Behavior-based Control: Examples from Navigation, Learning, and Group Behavior." *Journal of Experimental and Theoretical Artificial Intelligence*, 9(1997). Special Issue on Software Architectures for Physical Agents, H Hexmoor, I. Horswill, and D. Kortenkamp, eds.

[Tambe 97] Millind Tambe. Towards Flexible Teamwork (extended abstract). In *Socially Intelligent Agents*, AAAI technical report FS-97-02. AAAI Press, Meno Park, CA, 1997.

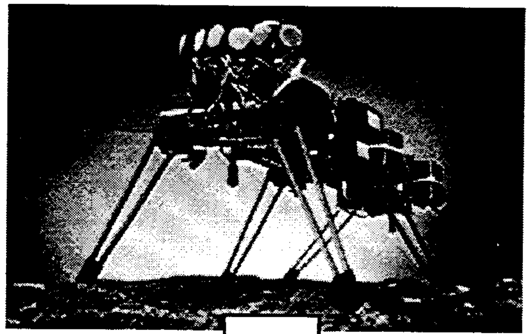
High performance reasoning on a low power budget

Dac Le
Ian Horswill

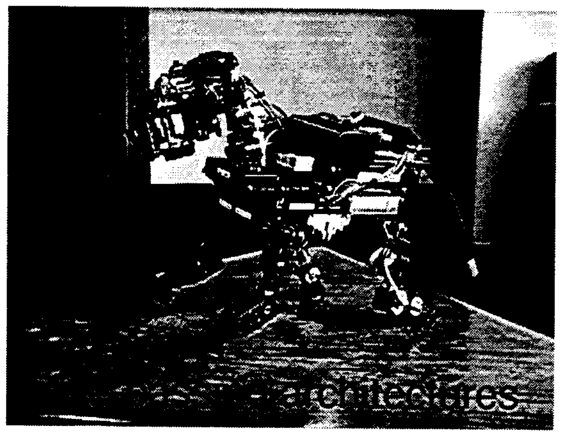
Northwestern University
Autonomous Mobile Robotics Group

Technology thrust: Integrating reasoning and reactivity

Behavior-based systems
- Low cost/low power
- Real-time response to contingencies



Symbolic/hybrid systems
- Declarative programming
- High level commands



Situational awareness

- Allocate sensory resources intelligently
- Sense task-relevant environmental changes
- Track task-impact in real time as environment changes

Can't take symbolic model for granted

Why is it hard?

Need to balance expressiveness and efficiency

- Symbolic reactive systems require ad-hoc code to update KB
  - ⇨ Often miss things
- Parallel reactive systems limited to propositional logic
  - ⇨ Behavior explosion

Approach

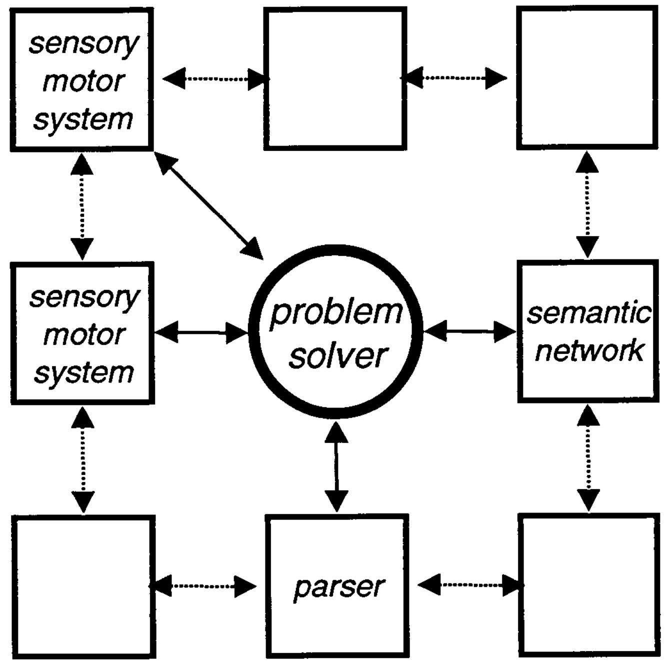
- *Multiple specialized representations*
  No uniform tree/tuple representation
- *Directly interface inference engine to sensors*
  Make sensors plug-compatible with KB
- *Compile inference rules to feed-forward network*
- *Rerun all inference rules, every clock, at sensor rates*

Impact

- Performance characteristics of behavior-based systems
  - Hard real-time response
  - Easy grounding in sensors
- Automatic control of attention
- Much more expressive representation language

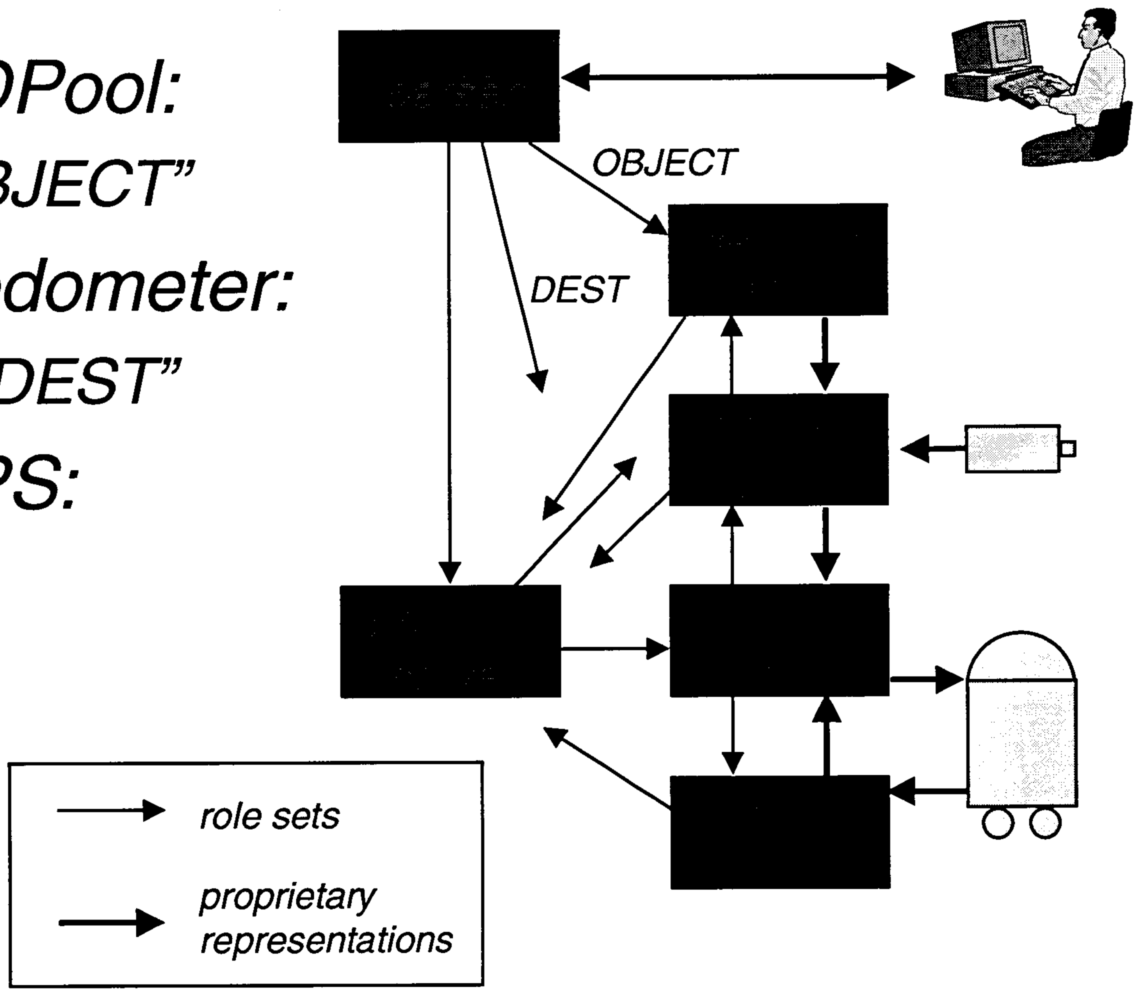
Role-passing interface contract

- Many experts glued together by problem solver
- Communication via role names
  - Action parameters
  - Predicate extensions
- Role ⇔ World bindings distributed through experts



"Bring the red ball here"

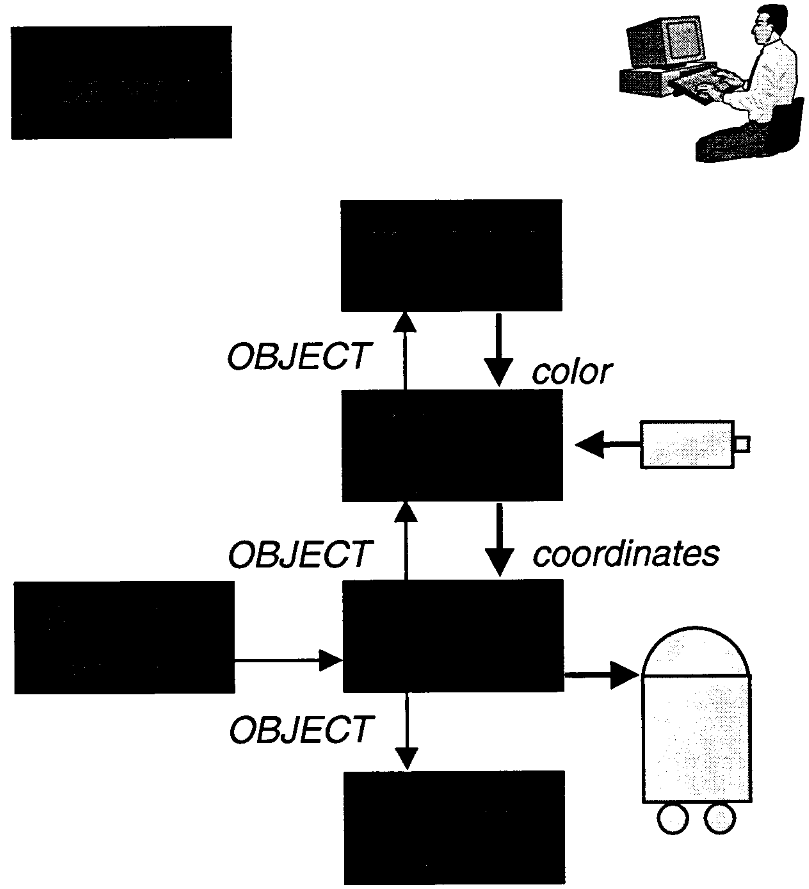
- **Parser to DPool:**
  Tag red "OBJECT"
- **Parser to odometer:**
  Tag home "DEST"
- **Parser to PS:**
  do BRING



Fetching a red ball

- PS to motor system:
  Approach OBJECT
- Motor system to trackers:
  Get coordinates of OBJECT
- Trackers to DPool:
  Get color of OBJECT

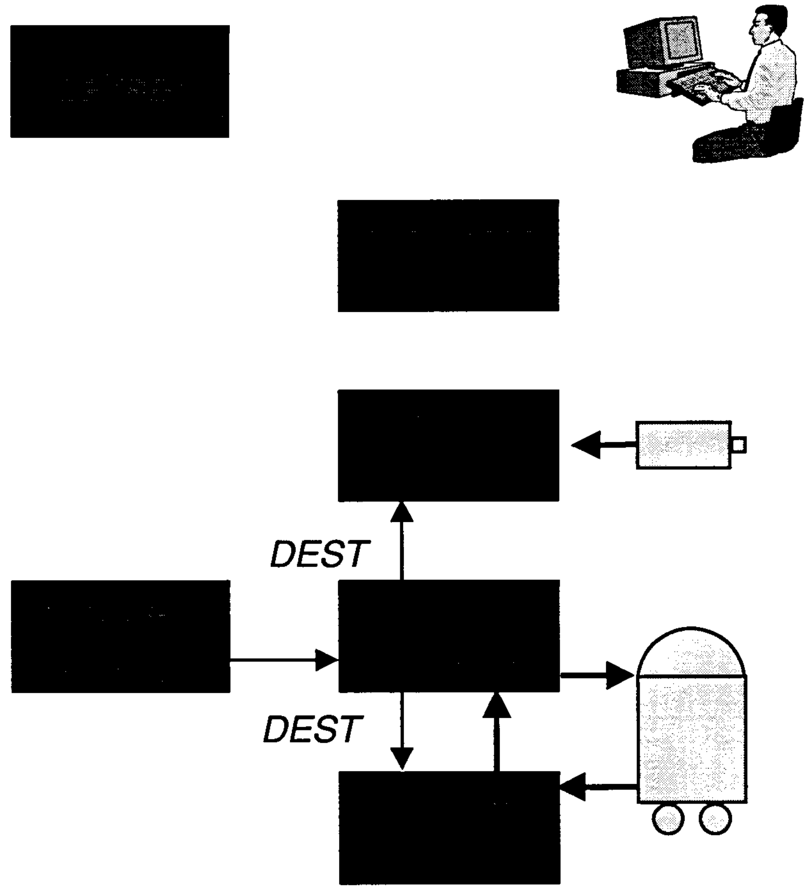
Robot drives to ball
Tracker reports OBJECT nearby
Robot grabs ball
Tracker reports OBJECT in hand



Fetching a red ball

- *PS to motor system:*
  Approach DEST
- *Motor system to odometers:*
  Get heading of DEST

Robot drives toward home
Odometer reports DEST nearby
Robot stops



Representing Predicates and Functions

Key idea: roles as indexicals

Predicate extensions can be represented with a single machine word or a narrow bus

	Source	Dest	Object	Agent
near(x)	T	F	F	T
see(x)	F	T	T	F

Function values require small vectors or busses

	Source	Dest	Object	Agent
distance(x)	15	0	0	17
direction(x)	14	87	35	-28

Computing Predicates and Functions

- *Sensory primitives generate extensions directly*
- *Derived functions and predicates computed using bitwise and/or:*

  $$\neg \ni x \,.\, P(x) \wedge (Q(x) \vee W(x))$$

  compiles to:

  ```
  (zerop (logand p (logior q w)))      ; lisp format
  !(p&(q|w))                           /* C format */
  ```

Propositional Attitudes

- *Knowledge as a valid bit*
- *Goal and know-goal bits*
  generated by problem solver
- *Automatic propagation*
  of valid bits, goal bits and know-goal bits

near(x)	*Source*	*Destination*	*Object*
Know	*F*	*T*	*T*
Goal	*F*	*F*	*T*
Know-goal	*T*	*F*	*T*

Microdoer fragment

*(define-pool-output-node **distance** tracker-distance tracker-pool)*
*(define-threshold-node **visually-near** < distance 13)*
*(define threshold-node **in-hand** < distance 3)*

Derived

*(define-disjunction **near** visually-near odometrically-near)*
*(define-disjunction **facing** visually-facing odometrically-facing)*

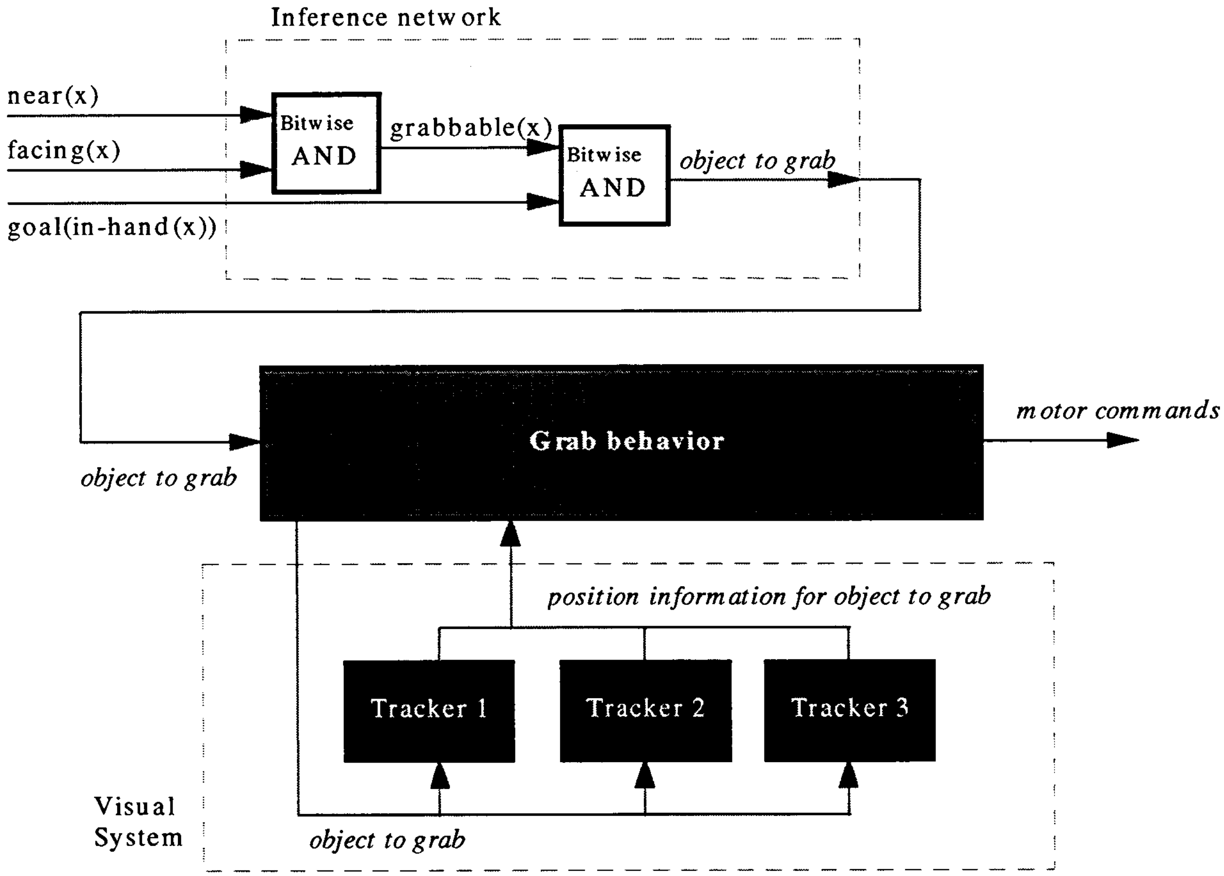
Regressed Conjunctions

*(define-serial-conjunction **grab** grabbable approach)*
*(define-parallel-conjunction **grabbable** near facing)*

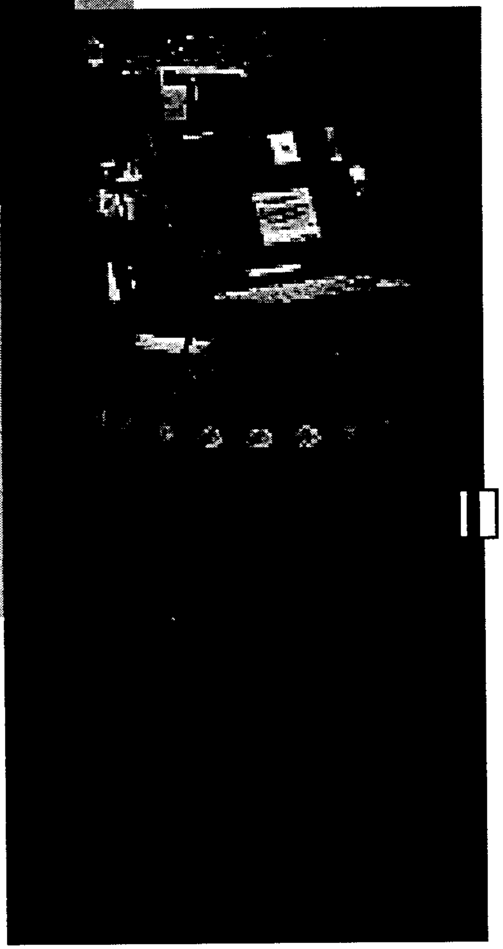
Explicit Rules

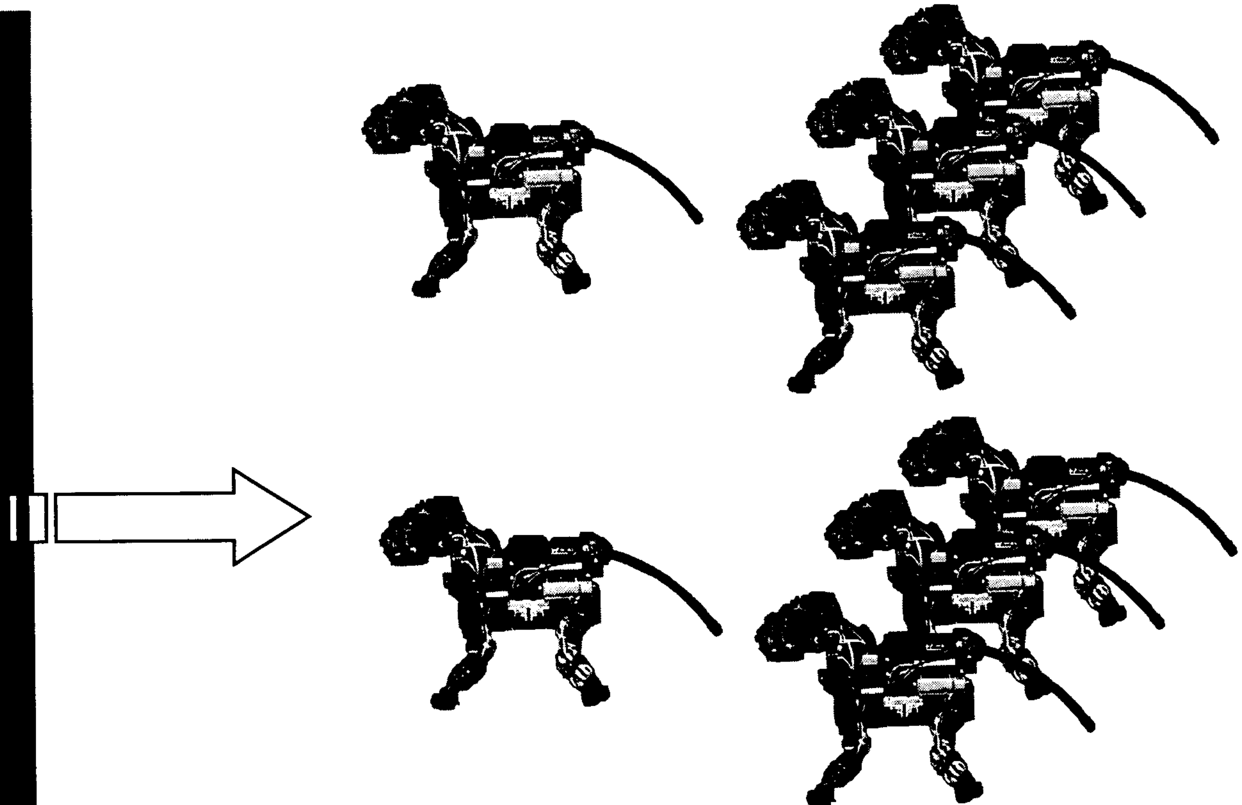
*(regress-unsatisfied **in-hand** grab)*
*(when-unsatisfied **near** follow-freespace!)*

Data flow


Inference network
near(x)
facing(x)
goal(in-hand(x))
Bitwise AND
grabbable(x)
Bitwise AND
object to grab
object to grab
Grab behavior
motor commands
position information for object to grab
Tracker 1
Tracker 2
Tracker 3
Visual System
object to grab

Technology thrust: Coordinated robot teams



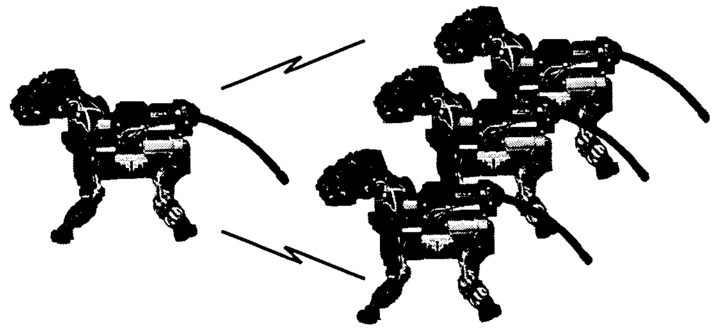


- Lower cost
- Redundancy
- Greater area coverage
- Situational control

Distributed role-passing

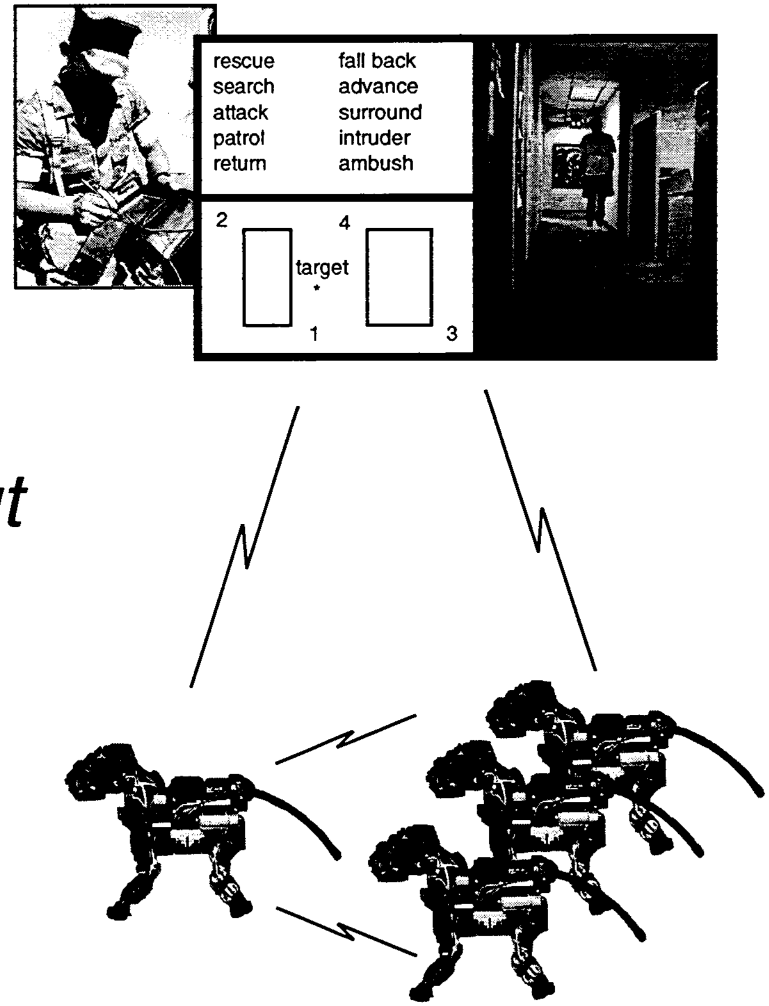
Use wireless network to:

- Extend distributed inference network beyond the individual
- Share situational awareness
- Coordinate squad-level operations



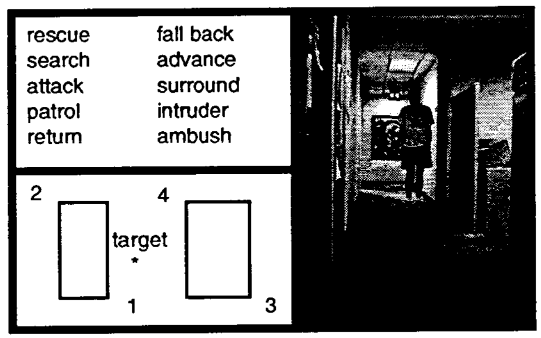
Command and control console

- Provides real-time display of
  - Telemetry
  - Situation assessment
  - Sensor data
- Natural language input of commands
- Implemented as role-passing robot w/o sensors



Application domain: Systematic spatial search

- Wide range of applications
  - Recon
  - Patrol
  - Rescue/disaster relief
  - Landmine clearance



- Difficult to implement with current technology
  - AI systems insufficiently responsive
  - B-Based systems can't easily represent it

Summary

KR language is restricted but...

- More expressive than B-Based systems
- Automatic update as world changes
- Hard real-time response
- Low power budget

  1000 inference rules, 100Hz update rate
  = 2MIPS = 2mW (using a StrongARM)
- Automatic control of attention